

Known algorithms for EDGE CLIQUE COVER are probably optimal

Marek Cygan *

Marcin Pilipczuk[†]Michał Pilipczuk[‡]

Abstract

In the EDGE CLIQUE COVER (ECC) problem, given a graph G and an integer k , we ask whether the edges of G can be covered with k complete subgraphs of G or, equivalently, whether G admits an intersection model on k -element universe. Gramm et al. [JEA 2008] have shown a set of simple rules that reduce the number of vertices of G to 2^k , and no algorithm is known with significantly better running time bound than a brute-force search on this reduced instance. In this paper we show that the approach of Gramm et al. is essentially optimal: we present a polynomial time algorithm that reduces an arbitrary 3-CNF-SAT formula with n variables and m clauses to an equivalent ECC instance (G, k) with $k = O(\log n)$ and $|V(G)| = O(n + m)$. Consequently, there is no $2^{2^{o(k)}} \text{poly}(n)$ time algorithm for the ECC problem, unless the Exponential Time Hypothesis fails. To the best of our knowledge, these are the first results for a natural, fixed-parameter tractable problem, and proving that a doubly-exponential dependency on the parameter is essentially necessary.

1 Introduction

Recently, there has been an increasing interest in not only improving the running times of exact algorithms for various NP-hard problems, but also in finding the limits of such improvements. *Parameterized complexity* is a very useful framework for such study: in this approach, an instance x of a parameterized problem comes with an integer parameter k . We say that a problem is *fixed parameter tractable (FPT)* if there exists an algorithm solving any instance (x, k) in time $f(k)|x|^{O(1)}$ for some computable function f . In other words, the exponential explosion of the running time, probably unavoidable for NP-hard problems, is encapsulated in a function of a parameter. With a wide range of possible parameters (e.g., intended solution size or structural graph parameters), the parameterized complexity paradigm allows much deeper insight into the hardness of NP-hard problems than the classical instance-size measure.

Although the definition of a fixed-parameter algorithm allows an arbitrarily fast growing function f , if f turns out to be relatively small (say, single-exponential), a fixed-parameter algorithm becomes practical for a reasonable range of values of the parameter. Therefore, since the dawn of parameterized complexity, researchers try to bound, as much as possible, the explosion of the running time hidden in the function f . In particular the goal of the part of parameterized complexity called by Marx [36] the *optimality program*, is to quantitatively understand what is the best possible f function in the running time.

In the last few years, this trend has been complemented by a research of lower bounds, usually conditional on the *Exponential Time Hypothesis* by Impagliazzo et al. [27]. Let c_k be the infimum over all positive reals c such that there exists an algorithm resolving satisfiability of n -variable k -CNF-SAT formulae in $O(2^{cn})$ time. The Exponential Time Hypothesis (ETH) asserts that $c_3 > 0$ (that is, 3-CNF-SAT

*IDSIA, University of Lugano, Switzerland, marek@idsia.ch

[†]Institute of Informatics, University of Warsaw, Poland, malcin@mimuw.edu.pl

[‡]Department of Informatics, University of Bergen, Norway, michal.pilipczuk@ii.uib.no

formulae cannot be resolved in subexponential time in the number of variables), whereas its stronger variant, *Strong Exponential Time Hypothesis* (SETH) [6, 26], asserts that $\lim_{k \rightarrow \infty} c_k = 1$ (in particular, an exhaustive search is the best possible solution for an arbitrary CNF formula).

Since the seminal paper of Impagliazzo et al. [27], many ETH-based lower bounds were developed (e.g., [16, 35, 34]), with the prominent example of tight bounds for $W[1]$ -hard problems [9, 10]. The most standard usage of ETH is to refute an existence of a subexponential algorithm for some problem by showing a linear (in the number of variables and clauses) reduction from an arbitrary 3-CNF-SAT formula and using to the so-called *Sparsification Lemma* [27]. In order to prove lower bounds for different complexities (different functions f , say $f(k) = 2^{O(k \log k)}$), the allowed dependency of k on n and m is more involved. Moreover, note that an ETH-based lower bound may exclude a subexponential algorithm, but says nothing about the possible base of the exponent in an $O(c^k |x|^{O(1)})$ fixed-parameter algorithm. For such lower bounds, one needs to assume SETH and either make the parameter independent of the number of clauses, or carefully use *Sparsification Lemma* as in some reductions of [11].

In 2011, two important results of by Lokshtanov, Marx and Saurabh [30, 31] show how to overcome the aforementioned difficulties: they developed a lower bound framework for proving ETH-based slightly superexponential lower bounds [31] as well as show that some core dynamic programming routines on graphs of bounded treewidth are optimal assuming SETH [30]. A significant contribution of these results is the development of new, sophisticated gadgets, which were found to be very inspiring. For example, both research directions initiated by Lokshtanov, Marx and Saurabh lead to settling down the tight bounds for algorithms for connectivity problems on graphs of bounded treewidth [13].

In this paper we make one step further in the quest of finding tight runtime bounds for parameterized problems by presenting (to the best of our knowledge) the first known double-exponential lower bound. Our problem of interest is an important combinatorial problem called **EDGE CLIQUE COVER**.

EDGE CLIQUE COVER

Input: An undirected graph G and an integer k .

Question: Does there exist a set of k subgraphs of G , such that each subgraph is a clique and each edge of G is contained in at least one of these subgraphs?

EDGE CLIQUE COVER is known to be NP-complete even in very restricted graph classes [8, 24, 37] and was widely studied under a few different names: **COVERING BY CLIQUES** (GT17), **INTERSECTION GRAPH BASIS** (GT59) [18] and **KEYWORD CONFLICT** [28]. It is known that the EDGE CLIQUE COVER problem is equivalent to the problem of finding a representation of a graph G as an intersection model with at most k elements in the universe [15, 22, 40]. Therefore a covering of a complex real-world network by a small number of cliques may reveal its hidden structure [23]. Further applications of EDGE CLIQUE COVER can be found in such various areas as computational geometry [1], applied statistics [20, 38], and compiler optimization [39]. Due to its importance, the EDGE CLIQUE COVER problem was studied from various perspectives, including approximation upper and lower bounds [2, 32], heuristics [3, 20, 28, 29, 38, 39] and polynomial-time algorithms for special graph classes [24, 25, 33, 37].

From the point of view of exact algorithms, a natural parameterization of EDGE CLIQUE COVER by the number of cliques was studied by Gramm et al. [19]. The authors propose a set of simple rules that reduce the number of vertices of the input graph to 2^k (the so-called *kernel*). Currently the best known fixed-parameter algorithm for EDGE CLIQUE COVER parameterized by k is a brute-force search on the 2^k -vertex kernel, which runs in double-exponential time in terms of k . Due to the importance of the EDGE CLIQUE COVER problem on one hand, and the lack of any improvement upon the very simple approach of Gramm et al. [19] on the other hand, EDGE CLIQUE COVER became a seasoned veteran of open problem sessions (with the most recent occurrence on the last Workshop on Kernels in Vienna, 2011). Only very

recently, a superset of the current authors [12] have shown that EDGE CLIQUE COVER is compositional, refuting (under the assumption $\text{NP} \not\subseteq \text{coNP/poly}$) an existence of a polynomial-time algorithm that reduces the size of the input graph to polynomial in k (the so-called *polynomial kernel*).

In this paper we complete the picture of the parameterized complexity of EDGE CLIQUE COVER. Our main technical result is the following reduction.

Theorem 1. *There exists a polynomial-time algorithm that, given a 3-CNF-SAT formula with n variables and m clauses, constructs an equivalent EDGE CLIQUE COVER instance (G, k) with $k = O(\log n)$ and $|V(G)| = O(n + m)$.*

The above theorem, together with a well-known result that an existence of a subexponential (in the number of variables and clauses) algorithm for verifying satisfiability of 3-CNF formulae violates ETH [27], proves the following lower bound.

Corollary 2. *Unless ETH fails, there does not exist an algorithm solving EDGE CLIQUE COVER in time $O(2^{2^{o(k)}} |V(G)|^{O(1)})$.*

We note that ETH is not necessary to refute existence of single-exponential algorithms for the problem.

Corollary 3. *Unless all problems in NP are solvable in quasipolynomial time, there does not exist an algorithm solving EDGE CLIQUE COVER in time $O(2^{k^{O(1)}} |V(G)|^{O(1)})$.*

Note that EDGE CLIQUE COVER, as a covering problem, can be solved by a dynamic programming algorithm in time $2^{O(|E(G)| + |V(G)|)}$. Thus, our result shows also the kernelization hardness for EDGE CLIQUE COVER.

Corollary 4. *Unless ETH fails, there does not exist a polynomial-time algorithm that, given an EDGE CLIQUE COVER instance (G, k) , outputs an equivalent instance (G', k') with $|V(G')| + |E(G')|$ bounded by $2^{o(k)}$.*

Moreover, note that if we would like to refute a polynomial kernel for EDGE CLIQUE COVER, we need significantly weaker assumption than ETH.

Corollary 5. *Unless all problems in NP are solvable in quasipolynomial time, there does not exist a polynomial-time algorithm that, given an EDGE CLIQUE COVER instance (G, k) , outputs an equivalent instance (G', k') with $|V(G')| + |E(G')|$ bounded polynomially in k .*

Finally, our reduction shows, that a polynomial time preprocessing routine reducing any instance of (G, k) of EDGE CLIQUE COVER to size $2^{o(k)}$, would make it possible to compress any r -CNF-SAT instance to $n^{o(r)}$ bits, which is not possible unless $\text{NP} \subseteq \text{coNP/poly}$, as observed by Dell and van Melkebeek [14]. Even more generally, we can rule out polynomial time compression to any language L with the output having size $2^{o(k)}$.

Corollary 6. *Unless $\text{NP} \subseteq \text{coNP/poly}$, there does not exist a polynomial-time algorithm that, given an EDGE CLIQUE COVER instance (G, k) , outputs an equivalent instance $A(G, k) \in \Sigma^*$ of some language L (i.e. $A(G, k) \in L$ iff (G, k) is a YES-instance) with $|A(G, k)| = 2^{o(k)}$.*

Proof. Consider a formula of r -CNF-SAT ϕ with n variables and m clauses. W.l.o.g. we may assume, that $m = O((2n)^r)$, since otherwise we may remove repeated clauses. Our first step is the standard reduction from r -CNF-SAT to 3-CNF-SAT, which in polynomial time outputs a formula ϕ' with $O(rm) = O(r(2n)^r)$

variables and $O(rm)$ clauses. Next, we use our reduction from Theorem 1, to create an instance (G, k) of EDGE CLIQUE COVER with $k = O(r \log n)$ and $|V(G)| = O(rm)$. Observe, that having a compression routine for EDGE CLIQUE COVER with $|A(G, k)|$ bounded by $2^{o(k)}$ would make it possible to express the initial formula ϕ with $n^{o(r)}$ bits, which is not possible due to Dell and van Melkebeek [14]. \square

To the best of our knowledge, the assumption $\text{NP} \not\subseteq \text{QP}$ and the assumption $\text{NP} \not\subseteq \text{coNP/poly}$, used (apart from Corollary 6) to refute a polynomial kernel for EDGE CLIQUE COVER in [12], are not known to be comparable. We are not aware of any other than Corollary 5 polynomial kernelization hardness result for fixed-parameter tractable problem outside the framework of compositions (for more on compositions, see e.g. [4, 5, 14, 17]).

Throughout the paper we investigate the graph we denote as H_ℓ , which is isomorphic to a clique on 2^ℓ vertices with a perfect matching removed, called the cocktail party graph. The core idea of the proof of Theorem 1 is the observation that a graph H_ℓ is a hard instance for the EDGE CLIQUE COVER problem, at least from the point of view of the currently known algorithms. Such a graph, while being immune to the reductions of Gramm et al. [19], can be quite easily covered with 2ℓ cliques, and there are multiple solutions of such size. Moreover, it is non-trivial to construct smaller clique covers for H_ℓ (but they exist).

In fact, the optimum size of a clique cover of cocktail party graphs with $2n$ vertices is proved to be $\min(k : n \leq \binom{k-1}{\lfloor k/2 \rfloor})$ for all $n > 1$ by Gregory and Pullman [21]. Moreover Chang et al. study cocktail party graphs in terms of rankwidth, which they prove to be unbounded in case of edge clique graphs of cocktail party graphs [7] (we refer to their work for appropriate definitions).

Acknowledgements. We thank Leszek Kołodziejczyk for some enlightening discussions on the complexity assumptions mentioned in this paper. Moreover we thank Daniel Lokshtanov and Ton Kloks for helpful comments and discussions.

2 Double-exponential lower bound

This section is devoted to the proof of Theorem 1. We start by introducing some notation. For an undirected graph G , by $V(G)$ and $E(G)$ we denote its vertex- and edge-sets respectively. For a set $X \subseteq V(G)$, a subgraph induced by X is denoted by $G[X]$. For two sets $X, Y \subseteq V(G)$, the set $E(X, Y)$ contains all edges of G that have one endpoint in X and a second endpoint in Y . As in this section we talk mostly about cliques, we allow ourselves some shortcuts in notation. If X is a subgraph or a subset of vertices of G , we call X a *clique in G* if X or $G[X]$ is a complete graph. We also often identify a subgraph being a clique in G with its vertex set. Moreover, for a bit-string c , by \bar{c} we denote its bitwise negation.

Recall that a graph H_ℓ is defined as a clique on 2^ℓ vertices with a perfect matching removed. In Section 2.1 we analyze in details graphs H_ℓ . It turns out that there is a large family of clique covers of size 2ℓ , where the clique cover consists of pairs of cliques, whose vertex sets are complements. We refer to such pairs as to *clique twins* and a clique cover consisting of clique twins is a *twin clique cover*. In particular, given any clique C of size $2^{\ell-1}$ in the graph H_ℓ , we can construct a twin clique cover of H_ℓ that contains C . Note that we have $2^{2^{\ell-1}}$ such starting cliques C in the graph H_ℓ : for each edge of the removed perfect matching, we choose exactly one endpoint into the clique. In our construction $\ell = \theta(\log n)$ and this clique C encodes the assignment of the variables in the input 3-CNF-SAT formula.

Section 2.2 contains the details of our construction. We construct a graph G with edge set $E(G)$ partitioned into two sets E^{imp} and E^{free} . The first set contains the important edges, that is, the ones that are

nontrivial to cover and play important role in the construction. The second set contains edges that are covered for free; in the end we add simplicial vertices (i.e., with neighbourhood being a clique) to the graph G to cover E^{free} . Note that without loss of generality we can assume that a closed neighbourhood of a non-isolated simplicial vertex is included in any optimal clique cover; however, we need to take care that we do not cover any edge of E^{imp} with such a clique and that we use only $O(\log n)$ simplicial vertices (as each such vertex adds a clique to the solution).

While presenting the construction in Section 2.2, we give informal explanations of the role of each gadget. In Section 2.3 we show formally how to translate a satisfying assignment of the input formula into a clique cover of the constructed graph, whereas the reverse translation is provided in Section 2.4.

2.1 Cocktail party graph

In this section we analyze graphs H_ℓ known as cocktail party graphs; see Figure 1 for an illustration for small values of ℓ . Recall that for an integer $\ell \geq 1$ the graph H_ℓ is defined as a complete graph on 2^ℓ vertices with a perfect matching removed. Note that a maximum clique in H_ℓ has $2^{\ell-1}$ vertices (i.e., half of all the vertices), and contains exactly one endpoint of each non-edge of H_ℓ . Moreover, if C is a maximum clique in H_ℓ , so is its complement $V(H_\ell) \setminus C$. This motivates the following definition.

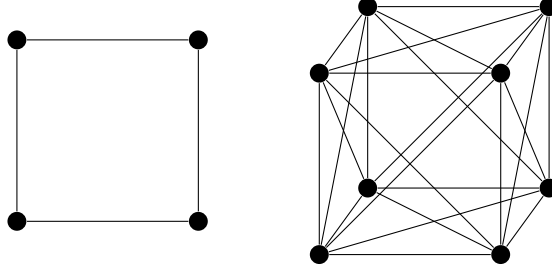


Figure 1: The graphs H_ℓ for $\ell = 2$ and $\ell = 3$. In the first case the optimum clique cover contains four two-vertex cliques and is a twin clique cover. In the second case an example twin clique cover is the set of all six faces of the cube; however, there exists a non-twin clique cover of H_3 of size five.

Definition 7. A pair of maximum cliques C and $V(H_\ell) \setminus C$ in H_ℓ is called *clique twins*. A clique cover of H_ℓ that consists of clique twins is called a *twin clique cover*.

The following lemma describes structure of twin clique covers of H_ℓ of size 2ℓ (i.e., containing ℓ twins).

Lemma 8. Assume we are given a set \mathcal{C}_0 of $1 \leq \delta \leq \ell$ clique twins with the following property: if we choose one clique from each of the δ clique twins, the intersection of the vertex sets of these cliques has size exactly $2^{\ell-\delta}$. Then there exists a twin clique cover \mathcal{C} of size 2ℓ that contains \mathcal{C}_0 .

Proof. We arbitrarily number the clique twins from \mathcal{C}_0 with numbers from 1 to δ , and in each clique twin we distinguish one clique labeled 0 and one labeled 1. Then, to each vertex $v \in V(H_\ell)$, we assign a δ -bit string \mathbf{c}_v that on a position γ ($1 \leq \gamma \leq \delta$) contains the bit assigned to the clique that contains v from the γ -th clique twins. Note that, as all subgraphs in \mathcal{C}_0 are cliques, for any non-edge uv of H_ℓ , the strings \mathbf{c}_u and \mathbf{c}_v are bitwise negations.

Fix a δ -bit string \mathbf{c} and consider $X_{\mathbf{c}} = \{v \in V(H_\ell) : \mathbf{c}_v = \mathbf{c}\}$. Note that any clique in \mathcal{C}_0 contains entire $X_{\mathbf{c}}$ or entire $X_{\bar{\mathbf{c}}}$ and, by the assumptions of the lemma, $|X_{\mathbf{c}}| = 2^{\ell-\delta}$. Moreover, as for a non-edge uv the

strings \mathbf{c}_u and \mathbf{c}_v are bitwise negations, each non-edge of H_ℓ connects a vertex from $X_{\mathbf{c}}$, for some \mathbf{c} , with a vertex from $X_{\bar{\mathbf{c}}}$. We can now label each vertex $v \in X_{\mathbf{c}}$ with bit string \mathbf{c}'_v of length $(\ell - \delta)$, such that in $X_{\mathbf{c}}$ each vertex receives a different label, and if uv is a non-edge of H_ℓ , then $\mathbf{c}'_u = \overline{\mathbf{c}'_v}$. In this manner each vertex $v \in V(H_\ell)$ receives a unique ℓ -bit label $\mathbf{c}_v \mathbf{c}'_v$ and for any non-edge uv of H_ℓ we have $\mathbf{c}_u \mathbf{c}'_u = \overline{\mathbf{c}_v \mathbf{c}'_v}$.

For an integer $1 \leq \gamma \leq \ell$ and a bit $c \in \{0, 1\}$, consider a set $C_{\gamma, c}$ consisting of those vertices of H_ℓ whose aforementioned ℓ -bit labels have γ -th bit set to c . As in H_ℓ a vertex v is connected with all other vertices except the one labeled with the bitwise negation of the label of v , $C_{\gamma, c}$ induces a clique. Moreover, for any edge $uv \in E(H_\ell)$, the labels of u and v agree on at least one bit, and the corresponding clique $C_{\gamma, c}$ contains the edge uv . As $C_{\gamma, 0} = V(H_\ell) \setminus C_{\gamma, 1}$, we infer that the family $\mathcal{C} = \{C_{\gamma, c} : 1 \leq \gamma \leq \ell, c \in \{0, 1\}\}$ is a twin clique cover of H_ℓ of size 2ℓ . We finish the proof of the lemma by noticing that $\{C_{\gamma, c} : 1 \leq \gamma \leq \delta, c \in \{0, 1\}\} = \mathcal{C}_0$. \square

Note that the above lemma for $\delta = 1$ implies that for any maximum clique C in H_ℓ there exists a twin clique cover of size 2ℓ that contains C . The next lemma treats about optimum twin clique covers of H_ℓ .

Lemma 9. *Let \mathcal{C} be a clique cover of H_ℓ that contains at least $\ell - 1$ clique twins. Then $|\mathcal{C}| \geq 2\ell$ and, if $|\mathcal{C}| = 2\ell$, then \mathcal{C} is a twin clique cover of H_ℓ .*

Proof. Let $\mathcal{C}_0 \subseteq \mathcal{C}$ be a set of $2\ell - 2$ cliques that form the assumed $\ell - 1$ clique twins. We use the family \mathcal{C}_0 to label the vertices of H_ℓ with $(\ell - 1)$ -bit labels similarly as in the proof of Lemma 8. That is, we arbitrarily number these clique twins with numbers 1 to $\ell - 1$, and in each clique twin we distinguish one clique labeled 0 and one labeled 1; a string \mathbf{c}_v for $v \in V(H_\ell)$ consists of $(\ell - 1)$ bits assigned to the cliques containing v . Again, for any non-edge uv of H_ℓ , the strings \mathbf{c}_u and \mathbf{c}_v are bitwise negations.

Fix a $(\ell - 1)$ -bit string \mathbf{c} and consider $X_{\mathbf{c}} = \{v \in V(H_\ell) : \mathbf{c}_v = \mathbf{c}\}$. Note that any clique in \mathcal{C}_0 contains entire $X_{\mathbf{c}}$ or entire $X_{\bar{\mathbf{c}}}$ and thus no clique in \mathcal{C}_0 covers the edges of $E(X_{\mathbf{c}}, X_{\bar{\mathbf{c}}})$. Moreover, as for any non-edge uv we have $\mathbf{c}_u = \overline{\mathbf{c}_v}$, the sets $X_{\mathbf{c}}$ and $X_{\bar{\mathbf{c}}}$ are of equal size.

As \mathcal{C} is a clique cover of H_ℓ , $\mathcal{C} \setminus \mathcal{C}_0$ covers $E(X_{\mathbf{c}}, X_{\bar{\mathbf{c}}})$. As $H_\ell[X_{\mathbf{c}} \cup X_{\bar{\mathbf{c}}}]$ is isomorphic to $K_{2|X_{\mathbf{c}}|}$ with a perfect matching removed, a direct check shows that if $|X_{\mathbf{c}}| \geq 3$ then $|\mathcal{C} \setminus \mathcal{C}_0| \geq 3$, that is, we need at least three cliques to cover $E(X_{\mathbf{c}}, X_{\bar{\mathbf{c}}})$. Thus, if $|\mathcal{C}| \leq 2\ell$, for each string \mathbf{c} we have $|X_{\mathbf{c}}| \leq 2$. As there are $2^{\ell-1}$ bit strings \mathbf{c} , and 2^ℓ vertices of H_ℓ , we infer that in this case $|X_{\mathbf{c}}| = 2$ for each bit string \mathbf{c} .

Fix a bit string \mathbf{c} . If $|X_{\mathbf{c}}| = 2$, then the graph $H_\ell[X_{\mathbf{c}} \cup X_{\bar{\mathbf{c}}}]$ is isomorphic to a 4-cycle and $E(X_{\mathbf{c}}, X_{\bar{\mathbf{c}}})$ contains two opposite edges of this cycle. These edges cannot be covered with a single clique. We infer that $|\mathcal{C} \setminus \mathcal{C}_0| \geq 2$, i.e., $|\mathcal{C}| \geq 2\ell$. Assume now that $|\mathcal{C}| = 2\ell$ and let $\mathcal{C} \setminus \mathcal{C}_0 = \{C, C'\}$. Note that for any bit string \mathbf{c} the clique C contains both endpoints of one edge of $E(X_{\mathbf{c}}, X_{\bar{\mathbf{c}}})$, and C' contains the endpoints of the second edge. Therefore $C = V(H_\ell) \setminus C'$ and the lemma is proven. \square

Let us remark that Lemma 9 implies that one cannot cover the graph H_ℓ with less than ℓ clique twins, i.e., the bound given by Lemma 8. Indeed, assume that there exists a cover of H_ℓ using $\ell' < \ell$ clique twins. If necessary, copy some of the clique twins in order to obtain a cover that uses exactly $\ell - 1$ twins. However, from Lemma 9 we infer that this cover needs to contain in fact more cliques, a contradiction.

2.2 Construction

Recall that, given a 3-CNF-SAT formula Φ , we are to construct an equivalent EDGE CLIQUE COVER instance with the number of cliques bounded logarithmically in the number of variables of Φ . We start with an empty graph G , and we subsequently add new gadgets to G . Recall that the edge set of G is partitioned into E^{free} and E^{imp} ; at the end of this section we show how to cover the set E^{free} with a small number of cliques,

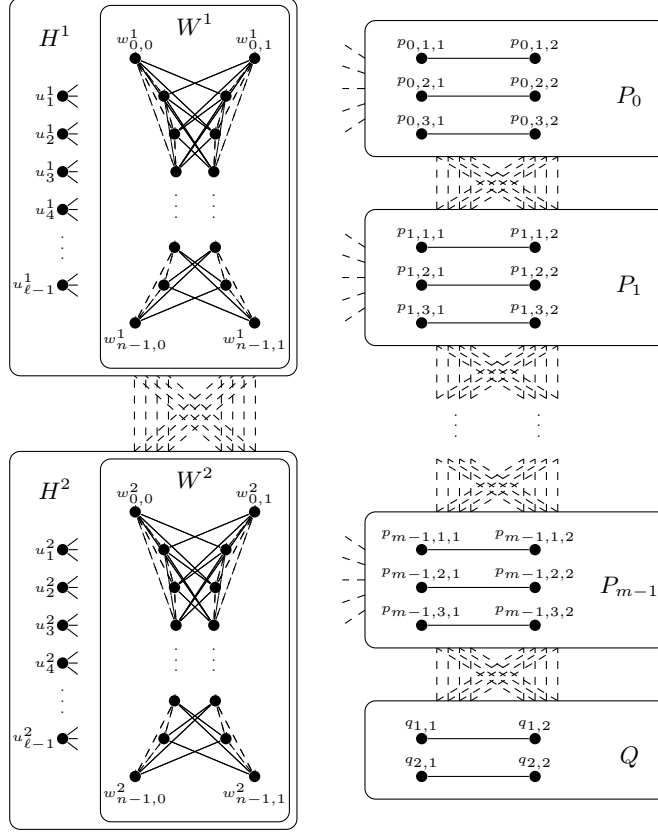


Figure 2: Illustration of the construction of the graph G . Solid edges belong to the set E^{imp} and dashed edges to E^{free} . The simplicial vertices are not illustrated, nor are the details which edges between the clause gadgets P_j and the assignment gadgets H^n are present in the graph G . Moreover not all edges of E^{free} between the vertices of gadgets Q and P_j are shown.

each induced by a closed neighbourhood of a simplicial vertex. We refer to Figure 2 for an illustration of the construction.

2.2.1 Preprocessing of the formula Φ

Let Vars denote the set of variables of Φ . By standard arguments, we can assume that in Φ each clause consists of exactly 3 literals, these literals contain different variables and no clause appears more than once. Moreover, we perform the following two regularization operations on Φ . First, we introduce some dummy variables into Vars , the set of variables of Φ , so that the number of variables is a power of two, and that there exists at least one dummy variable (i.e., the variable that does not appear in any clause). This operation at most doubles the number of variables in Vars . Second, we ensure that if Φ is satisfiable, then there exists a satisfying assignment of Φ that assigns true to exactly half of the variables, and false to the other half. This can be done by transforming Φ into $\Phi' = \Phi \wedge \bar{\Phi}$, where $\bar{\Phi}$ is a copy of Φ on duplicated set of variables $\bar{\text{Vars}}$ and, moreover, all the literals in $\bar{\Phi}$ are replaced with their negations. Clearly, if Φ' is satisfiable then the same assignment satisfies Φ in particular. Moreover, note that any satisfying assignment of Φ can be extended to a satisfying assignment of Φ' by assigning each copy of a variable the negation of the value of the original; this assignment assigns true to exactly half of the variables, and false to the other half. Observe

that the second operation does not change the properties ensured by the first operation, as it exactly doubles the number of variables. Moreover, in the satisfying assignment we can fix value of one dummy variable in Vars .

After performing the described operations, let n be the number of variables in Vars , m be the number of clauses of Φ , and $n = 2^\ell$. Note that $m = O(n^3)$ and $\log m = O(\log n) = O(\ell)$.

2.2.2 Assignment-encoding gadget

We assume that $\text{Vars} = \{x_0, x_1, \dots, x_{n-1}\}$ and that the 0-th variable is a dummy one (it serves in the construction as a true pattern). Take a graph H isomorphic to $H_{\ell+1}$ and denote its vertices by $w_{i,c}$ for $0 \leq i < n$, $c \in \{0, 1\}$; the non-edges of $H_{\ell+1}$ are $\{w_{i,0}w_{i,1} : 0 \leq i < n\}$. Let $W_c = \{w_{i,c} : 0 \leq i < n\}$ and $W = W_0 \cup W_1$. We put the edges of $H[W_0]$ and $H[W_1]$ into E^{free} and $E(W_0, W_1)$ into E^{imp} .

Moreover, we add $(\ell - 1)$ vertices u_γ , $1 \leq \gamma < \ell$, to the graph H . Each vertex u_γ is connected to all vertices of W via edges belonging to E^{imp} . This finishes the description of the assignment-encoding gadget H . We add two copies of the gadget H to the graph G , and denote the vertices of the η -th copy ($\eta \in \{1, 2\}$) by $w_{i,c}^\eta$ and u_γ^η . We define W^η , W_0^η , W_1^η and H^η in the natural way. In the graph G , for all indices $(i, c, i', c') \in (\{0, 1, \dots, n-1\} \times \{0, 1\})^2$ we connect each pair of vertices $w_{i,c}^1$ and $w_{i',c'}^2$ with an edge from E^{free} , i.e., we introduce a complete bipartite graph with edges belonging to E^{free} between W^1 and W^2 .

Let us now describe the intuition behind this construction. In the gadget H , the neighbourhood of each vertex u_γ is not a clique, thus any clique cover of H needs to include at least two cliques that contain u_γ . However, if we are allowed to use only two cliques per vertex u_γ , these cliques need to induce clique twins in the subgraph $H_{\ell+1}$ of H . With the assumption of only two cliques per vertex u_γ , the set of $(\ell - 1)$ vertices u_γ ensures that when covering $H_{\ell+1}$ we use ℓ clique twins: $(\ell - 1)$ from the vertices u_γ and one given by the edges in E^{free} (cliques W_0 and W_1). Lemma 9 asserts that the optimal way to complete a clique cover of $H_{\ell+1}$ is to use one more pair of clique twins: this clique twins, called *the assignment clique twins*, encode the assignment (and, as they are not bounded by the vertices u_γ , they can be used to verify the assignment in the forthcoming gadgets). Finally, we need two copies of the gadget H , as in the soundness proof we have one free clique that spoils the aforementioned argument; however, as the vertices $\{u_\gamma^\eta : 1 \leq \gamma < \ell, \eta \in \{1, 2\}\}$ form an independent set, it cannot spoil it in both copies at once. The edges between the sets W in the copies allow us to use the same two cliques as the assignment clique twins in both copies of the gadget H .

One could ask why we put edges from $H[W_0]$ and $H[W_1]$ into E^{free} , since in the previous section we have assumed that all the edges of H_ℓ are to be covered. The reason for this is that additional cliques with several vertices from W_0 or W_1 will appear, in order to cover other edges of E^{free} and for this reason we need to put edges from $H[W_0]$ and $H[W_1]$ into E^{free} and carefully investigate cliques that cover those edges.

2.2.3 Clause gadgets

We now introduce gadgets that verify correctness of the assignment encoded by the assignment clique twins, described in the previous paragraphs.

First, let us extend our notation. Let $\Phi = \Psi_0 \wedge \Psi_1 \wedge \dots \wedge \Psi_{m-1}$ and for integers j, α let $i(j, \alpha)$ be the index of the variable that appears in α -th literal in the clause Ψ_j . Moreover, let $c(j, \alpha) = 0$ if the α -th literal of Ψ_j is negative (i.e., $\neg x_{i(j, \alpha)}$) and $c(j, \alpha) = 1$ otherwise.

For each clause Ψ_j , we introduce into G a subgraph P_j isomorphic to $3K_2$, that is, $V(P_j) = \{p_{j, \alpha, \beta} : 1 \leq \alpha \leq 3, \beta \in \{1, 2\}\}$ and $E(P_j) = \{p_{j, \alpha, 1}p_{j, \alpha, 2} : 1 \leq \alpha \leq 3\}$. Moreover, we introduce into G a guard subgraph Q isomorphic to $2K_2$, that is, $V(Q) = \{q_{1,1}, q_{1,2}, q_{2,1}, q_{2,2}\}$ and $E(Q) = \{q_{1,1}q_{1,2}, q_{2,1}q_{2,2}\}$.

All the edges in all gadgets P_j and Q belong to E^{imp} . Moreover, we introduce the following edges to E^{free} . First, for each vertex of Q , we connect it with all vertices of all subgraphs P_j . Second, we connect each vertex $p_{j,\alpha,\beta}$ with all vertices $p_{j',\alpha',\beta'}$ for $j' \neq j$. Third, we connect each vertex $p_{j,\alpha,\beta}$ with all vertices in the sets W in both copies of the gadget H , except for $w_{0,1}^\eta$ and $w_{i(j,\alpha),1-c(j,\alpha)}^\eta$ for $\eta \in \{1, 2\}$. This finishes the description of the clause gadgets.

Let us now describe the intuition behind this construction. In each gadget P_j and in the guard subgraph Q the edges are independent, thus they need to be covered by different cliques. Two cliques are used to cover the edges of Q , and they can cover two out of three edges from each of the clause gadget P_j . The third one needs to be covered by the assignment clique twins from the gadgets H (as the gadgets P_j are not adjacent to the vertices u_γ^η), and it corresponds to the choice which literal satisfies the clause Ψ_j . The missing edge $p_{j,\alpha,\beta}x_{0,1}$ ensures that only one clique of the assignment clique twins is used to cover the edges of P_j . Finally, the missing edge $p_{j,\alpha,\beta}w_{i(j,\alpha),1-c(j,\alpha)}^\eta$ verifies that this clique encodes a satisfying assignment of Φ .

We note that it is non-trivial to cover the edges of E^{free} with $O(\ell) = O(\log n)$ cliques induced by closed neighbourhoods of simplicial vertices. This is done in the next sections by appropriately using bit-vectors.

2.2.4 Budget

We set the number of cliques to cover the edges of E^{imp} as $k_0 = 2 \cdot 2 \cdot (\ell - 1) + 2 + 2 = 4\ell$ — two for each vertex u_γ^η , two for the assignment clique twins in H , and two for the cliques that contain the edges of Q . The final number of cliques k is the sum of k_0 and the number of simplicial vertices introduced in the next section.

2.2.5 Covering the free edges

In this section we show that the edges of E^{free} can be covered by a small number of cliques, without accidentally covering any edge of E^{imp} . Moreover, such covering can be constructed in polynomial time. To construct the final EDGE CLIQUE COVER instance, for each clique of this clique cover we introduce a simplicial vertex adjacent to the vertex set of this clique, and raise the desired number of cliques by one.

Lemma 10. *The graph $G^{\text{free}} = (V(G), E^{\text{free}})$ admits a clique cover of size $46 + 36\lceil \log m \rceil + 24\ell = O(\log n)$. Moreover, such a clique cover can be constructed in polynomial time.*

Proof. We start by noting that the free edges in the copies of the gadget H , and between these copies, can be covered by four cliques: $\{w_{\gamma,c}^1 : 0 \leq \gamma < n\} \cup \{w_{\gamma,c'}^2 : 0 \leq \gamma < n\}$ for $(c, c') \in \{0, 1\} \times \{0, 1\}$. Similarly, the edges incident to the guard gadget Q can be covered with 24 cliques: $\{q_{\alpha,\beta}\} \cup \{p_{j,\alpha',\beta'} : 0 \leq j < m\}$ for $(\alpha, \beta, \alpha', \beta') \in \{1, 2\} \times \{1, 2\} \times \{1, 2, 3\} \times \{1, 2\}$.

Covering the edges between different gadgets P_j requires a bit more work. For each $1 \leq \gamma \leq \lceil \log m \rceil$ and $(\alpha, \beta, \alpha', \beta') \in (\{1, 2, 3\} \times \{1, 2\})^2$ we take a clique $C_{\gamma,\alpha,\beta,\alpha',\beta'}^P$ that contains exactly one vertex from each gadget P_j : if the γ -th bit of the binary representation of j equals 0, $p_{j,\alpha,\beta} \in C_{\gamma,\alpha,\beta,\alpha',\beta'}^P$, and otherwise $p_{j,\alpha',\beta'} \in C_{\gamma,\alpha,\beta,\alpha',\beta'}^P$. Clearly, $C_{\gamma,\alpha,\beta,\alpha',\beta'}^P$ induces a clique in G^{free} , as it contains exactly one vertex from each gadget P_j . Let us now verify that all edges between the gadgets P_j are covered by these $36\lceil \log m \rceil$ cliques. Take any edge $p_{j,\alpha,\beta}p_{j',\alpha',\beta'} \in E^{\text{free}}$, $j \neq j'$. Assume that the binary representations of j and j' differ on the γ -th bit; without loss of generality, assume that the γ -th bit of j is 0, and the γ -th bit of j' is 1. Then the clique $C_{\gamma,\alpha,\beta,\alpha',\beta'}^P$ contains both $p_{j,\alpha,\beta}$ and $p_{j',\alpha',\beta'}$.

We now handle the edges that connect the two copies of the gadget H with the gadgets P_j . First, we take care of the edges that are incident to the vertices $w_{0,0}^\eta$. This can be easily done with 6 cliques: for

each $(\alpha, \beta) \in \{1, 2, 3\} \times \{1, 2\}$ we take a clique that contains $w_{0,0}^1, w_{0,0}^2$ as well as all vertices $p_{j,\alpha,\beta}$ for $0 \leq j < m$. Second, we take care of the edges $p_{j,\alpha,\beta} w_{i(j,\alpha),c(j,\alpha)}^\eta$. To this end, we take 12 cliques: for each $(\alpha, \beta, c) \in \{1, 2, 3\} \times \{1, 2\} \times \{0, 1\}$ we take a clique that contains $w_{i,c}^\eta$ for $1 \leq i < n, \eta \in \{1, 2\}$ as well as all vertices $p_{j,\alpha,\beta}$ that satisfy $c(j, \alpha) = c$.

We are left with the edges of form $p_{j,\alpha,\beta} w_{i,c}^\eta$ for $i \notin \{0, i(j, \alpha)\}$. These edges can be covered in a similar fashion to the edges between the gadgets P_j . For each $1 \leq \gamma \leq \ell$ and $(\alpha, \beta, c, c') \in \{1, 2, 3\} \times \{1, 2\} \times \{0, 1\}^2$ we construct a clique $C_{\gamma,\alpha,\beta,c,c'}^W$ that contains all vertices $w_{i,c}^\eta$ for $\eta \in \{1, 2\}$ and $1 \leq i < n$ such that the γ -th bit of the binary representation of i equals c' , as well as all vertices $p_{j,\alpha,\beta}$ for $0 \leq j < m$ such that the γ -th bit of the binary representation of $i(j, \alpha)$ equals $1 - c'$. To see that $C_{\gamma,\alpha,\beta,c,c'}^W$ is indeed a clique in G^{free} , note that it contains only edges in $G[W_0^1 \cup W_0^2]$ or $G[W_1^1 \cup W_1^2]$, between different gadgets P_j , and edges of the form $p_{j,\alpha,\beta} w_{i,c}^\eta$ where $i \neq 0$ and $i \neq i(j, \alpha)$ (the indices i and $i(j, \alpha)$ must differ on the γ -th bit in the clique $C_{\gamma,\alpha,\beta,c,c'}^W$). We finish the proof of the lemma by verifying that all edges of the form $p_{j,\alpha,\beta} w_{i,c}^\eta$ for $i \notin \{0, i(j, \alpha)\}$ are covered by these 24ℓ cliques. As $i \neq i(j, \alpha)$, there exists $1 \leq \gamma \leq \ell$ such that i and $i(j, \alpha)$ differ on the γ -th bit of their binary representations. Let c' be the γ -th bit of the binary representation of i . We infer that both $p_{j,\alpha,\beta}$ and $w_{i,c}^\eta$ are included in the clique $C_{\gamma,\alpha,\beta,c,c'}^W$ and the lemma is proven. \square

Recall that for each clique constructed by Lemma 10 we add a simplicial vertex to G that is adjacent to all vertices of this clique. The simplicial vertices are independent in G . As discussed earlier, we can assume that for any non-isolated simplicial vertex s in G , any optimal clique cover in G contains a clique whose vertex set equals to the closed neighbourhood of s .

We conclude the construction section by setting the desired number of cliques k to be the sum of k_0 and the number of aforementioned simplicial vertices, $k = 4\ell + 46 + 36\lceil \log m \rceil + 24\ell = O(\log n)$.

2.3 Completeness

In this section we show how to translate a satisfying assignment of the input formula Φ into a clique cover of G of size k .

Lemma 11. *If the input formula Φ is satisfiable, then there exists a clique cover of the graph G of size k .*

Proof. Let $\phi : \{0, 1, \dots, n-1\} \rightarrow \{0, 1\}$ be a satisfying assignment of Φ , that is, $\phi(i)$ is the value of x_i , 0 stands for false and 1 stands for true. By the properties of the preprocessing step of the construction, we may assume that $|\phi^{-1}(0)| = |\phi^{-1}(1)| = |\text{Vars}|/2 = 2^{\ell-1}$ and that $\phi(0) = 0$ (as x_0 is a dummy variable).

We start the construction of the clique cover \mathcal{C} of the graph G by taking into \mathcal{C} , for each of the $46 + 36\lceil \log m \rceil + 24\ell$ simplicial vertices of G constructed in Lemma 10, a clique induced by the closed neighbourhood of the simplicial vertex. In this manner we cover all edges of E^{free} , and we are left with a budget of 4ℓ cliques.

We define the assignment clique twins C_0^A and C_1^A . For each clause Ψ_j of Φ , let $\alpha(j)$ be an index of a literal that is satisfied by ϕ in Ψ_j (if there is more than one such literal, we choose an arbitrary one). The clique C_0^A contains the vertices $w_{i,\phi(i)}^\eta$ for $0 \leq i < n$ and $\eta \in \{1, 2\}$ as well as the following vertices from the clause gadgets: $p_{j,\alpha(j),\beta}$ for $0 \leq j < m, \beta \in \{1, 2\}$. Note that C_0^A is indeed a clique, since the only missing edges between vertices $w_{i,c}^\eta$ and $p_{j,\alpha,\beta}$ are either incident to $w_{0,1}^\eta$ (but $\phi(0) = 0$) or of the form $w_{i(j,\alpha),1-c(j,\alpha)}^\eta p_{j,\alpha,\beta}$ (but $\phi(i(j, \alpha(j)))$ satisfies $\alpha(j)$ -th literal of Ψ_j , i.e., $c(j, \alpha(j)) = \phi(i(j, \alpha(j)))$).

The clique C_1^A is the twin (complement) of the clique C_0^A in both copies of the graph $H_{\ell+1}$, i.e., $C_1^A = \{w_{i,1-\phi(i)}^\eta : 0 \leq i < n, \eta \in \{1, 2\}\}$. Clearly, C_1^A is a clique in G .

Let us now fix $\eta \in \{1, 2\}$ and focus on the graph $G[W^\eta]$ isomorphic to $H_{\ell+1}$. The edges of E^{free} in this subgraph form clique twins $\{w_{i,c}^\eta : 0 \leq i < n\}$ for $c \in \{0, 1\}$. The assignment clique twins C_0^A and C_1^A form second clique twins in $G[W^\eta]$, after truncating them to this subgraph. Moreover, the assumption that ϕ evaluates exactly half of the variables to false and half to true implies that these two clique twins satisfy the assumptions of Lemma 8. We infer that all remaining edges of $G[W^\eta]$ can be covered by $\ell - 1$ clique twins; we add the vertex u_γ^η to both cliques of the γ -th clique twin, and add these clique twins to the constructed clique cover \mathcal{C} . In this manner we cover all edges incident to all vertices u_γ^η for $1 \leq \gamma < \ell$, $\eta \in \{1, 2\}$. As we perform this construction for both values $\eta \in \{1, 2\}$, we use $4\ell - 4$ cliques, and we are left with a budget of two cliques.

The cliques introduced in the previous paragraph cover all edges in E^{imp} in both copies of the assignment gadget H . We are left with the clause gadgets P_j and the guard gadget Q . Recall that the clique C_0^A covers one out of three edges in each gadget P_j . Thus it is straightforward to cover the remaining edges with two cliques: each clique contains both endpoints of exactly one uncovered edge from each gadget P_j and the gadget Q . This finishes the proof of the lemma. \square

2.4 Soundness

In this section we show a reverse transformation: a clique cover of G of size at most k cannot differ much from the one constructed in the proof of Lemma 11 and, therefore, encodes a satisfying assignment of the input formula Φ .

Lemma 12. *If there exists a clique cover of G of size at most k , then the input formula Φ is satisfiable.*

Proof. Let \mathcal{C} be a clique cover of size at most k of G . As G contains $k - 4\ell$ simplicial vertices, without loss of generality we may assume that, for each such simplicial vertex s , the family \mathcal{C} contains a clique induced by the closed neighbourhood of s . These cliques cover the edges of E^{free} , but no edge of E^{imp} . Let $\mathcal{C}_0 \subseteq \mathcal{C}$ be the set of the remaining cliques; $|\mathcal{C}_0| \leq 4\ell$.

Let us start with analyzing the guard gadget Q . It contains two independent edges from E^{imp} . Thus, \mathcal{C}_0 contains two cliques, each containing one edge of Q . Denote these cliques by C_1^Q and C_2^Q . Note that each clause gadget P_j contains three independent edges, and only two of them may be covered by the cliques C_1^Q and C_2^Q . Thus there exists at least one additional clique in \mathcal{C}_0 that contains an edge of at least one gadget P_j ; let us denote this clique by C_0^A (if there is more than one such clique, we choose an arbitrary one).

Each vertex u_γ^η for $1 \leq \gamma < \ell$, $\eta \in \{1, 2\}$ needs to be contained in at least two cliques of \mathcal{C}_0 , since all its incident edges are in E^{imp} and the neighbourhood of u_γ^η is not a clique. Moreover, no vertex u_γ^η may belong to C_1^Q , C_2^Q nor to C_0^A , as these vertices are not adjacent to the vertices of P_j and Q . As there are $2\ell - 2$ vertices u_γ^η , the vertices u_γ^η are independent, and $|\mathcal{C}_0 \setminus \{C_1^Q, C_2^Q, C_0^A\}| \leq 4\ell - 3$, we infer that at most one vertex u_γ^η may be contained in more than two cliques from \mathcal{C}_0 . Without loss of generality we can assume that this vertex belongs to the second copy of the assignment gadget H , that is, each vertex u_γ^1 for $1 \leq \gamma < \ell$ belongs to exactly two cliques $C_{\gamma,0}^U, C_{\gamma,1}^U \in \mathcal{C}_0$.

Note that the only way to cover the edges incident to u_γ^1 with only two cliques $C_{\gamma,0}^U, C_{\gamma,1}^U$ is to take these cliques to induce clique twins in $G[W^1] \cong H_{\ell+1}$. That is, $C_{\gamma,0}^U$ consists of u_γ^1 and exactly one endpoint of each non-edge of $G[W^1]$, and $C_{\gamma,1}^U = \{u_\gamma^1\} \cup (W^1 \setminus C_{\gamma,0}^U)$.

We infer that the cliques $\{C_{\gamma,c}^U : 1 \leq \gamma < \ell, c \in \{0, 1\}\}$, together with the clique twins formed by the edges of E^{free} in $G[W^1]$, sum up to ℓ clique twins in $G[W^1] \cong H_{\ell+1}$. Moreover, the edges of $G[W^1]$ may be covered with only two additional cliques (including C_0^A): there are at least $4\ell - 4$ cliques that contain vertices u_γ^1 or u_γ^2 , out of which exactly $2\ell - 2$ can contain vertices from W^1 , while at least two other cliques

have to contain vertices of Q , thus having to be disjoint with W^1 . Lemma 9 asserts that the only way to cover the edges of $G[W^1] \cong H_{\ell+1}$ is to use one additional pair of cliques that form a clique twin; thus, $C_0^A \cap W^1$ is a maximum clique in $G[W^1]$ and \mathcal{C}_0 contains a clique C_1^A such that $C_1^A \cap W^1 = W^1 \setminus C_0^A$.

Recall that the clique C_0^A contained an edge from at least one gadget P_j . Therefore, $x_{0,1}^1 \notin C_0^A$, as $x_{0,1}^1$ is not adjacent to any vertex in any gadget P_j . Since C_0^A and C_1^A induce clique twins in $G[W^1]$, we infer that $x_{0,1}^1 \in C_1^A$ and C_1^A is disjoint with all gadgets P_j . As the vertices u_γ^η are not adjacent to the gadgets P_j , the cliques that cover the edges incident to the vertices u_γ^η cannot cover any edges of the gadgets P_j either. We infer that the edges of the gadgets P_j are covered by only three cliques — C_0^A , C_1^Q and C_2^Q — and that in each gadget P_j , each of this three cliques contains exactly one edge. For a clause Ψ_j , let $\alpha(j)$ be the index of the literal whose edge is covered by C_0^A .

We claim that an assignment $\phi : \{0, 1, \dots, n-1\} \rightarrow \{0, 1\}$ that assigns the value $\phi(i)$ to the variable x_i in such a manner that $w_{i,\phi(i)}^1 \in C_0^A$, satisfies the formula Φ . More precisely, we claim that for each clause Ψ_j , the $\alpha(j)$ -th literal of Ψ_j satisfies this clause in the assignment ϕ . Indeed, as $p_{j,\alpha(j),\beta} \in C_0^A$ for $\beta \in \{1, 2\}$, and $p_{j,\alpha(j),\beta}$ is not adjacent to $w_{i(j,\alpha(j)),1-c(j,\alpha(j))}^1$, we have that $w_{i(j,\alpha(j)),c(j,\alpha(j))}^1 \in C_0^A$ and $\phi(i(j,\alpha(j))) = c(j,\alpha(j))$. This finishes the proof of the lemma and concludes the proof of Theorem 1. \square

3 Conclusions

In this paper we have shown a double-exponential lower bound for EDGE CLIQUE COVER parameterized by the number of cliques, obtaining tight complexity bounds for this problem.

Prior to this work, several matching lower bounds for $2^{O(k)} \text{poly}(n)$, $2^{O(\sqrt{k})} \text{poly}(n)$ and $2^{O(k \log k)} \text{poly}(n)$ time algorithms for parameterized problems were already known. However, to the best of our knowledge EDGE CLIQUE COVER is the first example of a natural parameterization for which double-exponential upper and lower bounds are proved. We hope our work will inspire further results of this type.

References

- [1] P. K. Agarwal, N. Alon, B. Aronov, and S. Suri. Can visibility graphs be represented compactly? *Discrete & Computational Geometry*, 12:347–365, 1994.
- [2] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 1999.
- [3] M. Behrisch and A. Taraz. Efficiently covering complex networks with cliques of similar vertices. *Theor. Comput. Sci.*, 355(1):37–47, 2006.
- [4] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.
- [5] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Cross-composition: A new technique for kernelization lower bounds. In *STACS*, pages 165–176, 2011.
- [6] C. Calabro, R. Impagliazzo, and R. Paturi. The complexity of satisfiability of small depth circuits. In *IWPEC*, pages 75–85, 2009.

- [7] M.-S. Chang, T. Kloks, and C.-H. Liu. Edge-clique graphs of cocktail parties have unbounded rankwidth. *CoRR*, abs/1205.2483, 2012.
- [8] M.-S. Chang and H. Müller. On the tree-degree of graphs. In *WG*, pages 44–54, 2001.
- [9] J. Chen, X. Huang, I. A. Kanj, and G. Xia. On the computational hardness based on linear fpt-reductions. *J. Comb. Optim.*, 11(2):231–247, 2006.
- [10] J. Chen, X. Huang, I. A. Kanj, and G. Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006.
- [11] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström. On problems as hard as CNF-SAT. In *IEEE Conference on Computational Complexity*, pages 74–84, 2012.
- [12] M. Cygan, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and M. Wahlström. Clique cover and graph separation: New incompressibility results. In *ICALP (I)*, pages 254–265, 2012.
- [13] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *FOCS*, pages 150–159, 2011.
- [14] H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *STOC*, pages 251–260, 2010.
- [15] P. Erdős, A. W. Goodman, and L. Posa. The representation of a graph by set intersections. *Canadian Journal of Mathematics*, 18:106–112, 1966.
- [16] F. V. Fomin, P. A. Golovach, D. Lokshtanov, and S. Saurabh. Algorithmic lower bounds for problems parameterized with clique-width. In *SODA*, pages 493–502, 2010.
- [17] L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.
- [18] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W.H. Freeman, 1979.
- [19] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Data reduction and exact algorithms for clique cover. *ACM Journal of Experimental Algorithmics*, 13, 2008.
- [20] J. Gramm, J. Guo, F. Hüffner, R. Niedermeier, H.-P. Piepho, and R. Schmid. Algorithms for compact letter displays: Comparison and evaluation. *Computational Statistics & Data Analysis*, 52(2):725–736, 2007.
- [21] D. A. Gregory and N. J. Pullman. On a clique covering problem of orlin. *Discrete Mathematics*, 41(1):97–99, 1982.
- [22] J. L. Gross and J. Yellen. *Graph Theory and its Applications*. CRC Press, 2006.
- [23] J.-L. Guillaume and M. Latapy. Bipartite structure of all complex networks. *Inf. Process. Lett.*, 90(5):215–221, 2004.

- [24] D. N. Hoover. Complexity of graph covering problems for graphs of low degree. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 11:187–208, 1992.
- [25] W.-L. Hsu and K.-H. Tsai. Linear time algorithms on circular-arc graphs. *Inf. Process. Lett.*, 40(3):123–129, 1991.
- [26] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.
- [27] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
- [28] E. Kellerman. Determination of keyword conflict. *IBM Technical Disclosure Bulletin*, 16(2):544–546, 1973.
- [29] L. T. Kou, L. J. Stockmeyer, and C.-K. Wong. Covering edges by cliques with regard to keyword conflicts and intersection graphs. *Communications of the ACM*, 21(2):135–139, 1978.
- [30] D. Lokshtanov, D. Marx, and S. Saurabh. Known algorithms on graphs on bounded treewidth are probably optimal. In *SODA*, pages 777–789, 2011.
- [31] D. Lokshtanov, D. Marx, and S. Saurabh. Slightly superexponential parameterized problems. In *SODA*, pages 760–776, 2011.
- [32] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.
- [33] S. Ma, W. D. Wallis, and J. Wu. Clique covering of chordal graphs. *Utilitas Mathematica*, 36:151–152, 1989.
- [34] D. Marx. On the optimality of planar and geometric approximation schemes. In *FOCS*, pages 338–348, 2007.
- [35] D. Marx. Can you beat treewidth? *Theory of Computing*, 6(1):85–112, 2010.
- [36] D. Marx. What’s next? future directions in parameterized complexity. In *The Multivariate Algorithmic Revolution and Beyond*, pages 469–496, 2012.
- [37] J. B. Orlin. Contentment in graph theory: Covering graphs with cliques. *Indagationes Mathematicae (Proceedings)*, 80(5):406–424, 1977.
- [38] H.-P. Piepho. An algorithm for a letter-based representation of all-pairwise comparisons. *Journal of Computational and Graphical Statistics*, 13(2):456–466, 2004.
- [39] S. Rajagopalan, M. Vachharajani, and S. Malik. Handling irregular ILP within conventional VLIW schedulers using artificial resource constraints. In *CASES*, pages 157–164, 2000.
- [40] F. S. Roberts. Applications of edge coverings by cliques. *Discrete Applied Mathematics*, 10(1):93 – 109, 1985.